



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant : John Robinson
Serial No. : 09/971,086
Filed : October 5, 2001
Title : REROUTING MEDIA TO SELECTED MEDIA APPLICATIONS

Art Unit : Unknown
Examiner : Unknown

Commissioner for Patents
Washington, D.C. 20231

PRELIMINARY AMENDMENT

Prior to examination, please amend the application as follows:

In the specification:

Replace the paragraph beginning at page 8, line 1 with the following rewritten paragraph:

-- IProtocolSink is an interface for pluggable MIME filters that is used to receive notifications if/when data is available for filtering. IProtocolSink allows the APP to obtain a window handle for the client performing the bind operation. Although this window handle typically is used for other purposes, it can be used to determine whether an instance of a MIME filter originated from an internal browser.--

Replace the paragraph beginning at page 9, line 6 with the following rewritten paragraph:

--If the data stream was requested by an instance of an internal browser, the registry structure is modified to include registrations of media players appropriate for internal instances of the browser or some other list of alternate media player registrations is accessed, and that modified or alternate registry is analyzed to determine which media player is registered for the MIME type (step 560). Then, an instance of an appropriate alternate media player is established (step 570) and the data stream is effectively routed for rendering to the alternate media player (step 580). Absent further processing, after the media stream is rendered, the MIME filter returns the data stream to URLMON. Therefore, to prevent invocation of a second media player according to the modified Windows[®] registry, it may be desirable or necessary to sink the data stream (step 590).--

Replace the paragraph beginning at page 9, line 16 with the following rewritten paragraph:

--Fig. 6 illustrates one implementation for sinking a call for a default media player ordinarily generated for data streams of similar MIME type by external instances of a browser (step 590). In this implementation, the MIME filter changes the MIME type of the data stream returned to URLMON to a predetermined MIME type (step 610) for which no media player or a dummy media player has been registered. As a result, when the browser detects the altered MIME type in the data stream, the browser will initiate either no media player or an instance of this dummy media player, as appropriate (step 620). Then, to complete the data sink, the modified data stream is effectively routed to this dummy media player, which accesses the data stream and either discards the data stream or performs some trivial or background function on the data stream.--

Replace the paragraph beginning at page 11, line 1 with the following rewritten paragraph:

--The browser then initiates an instance of the alternate media player (step 770) and the data stream is effectively routed to it for rendering (step 780).--

09/971,086 "9801/2650

Applicant : John Robinson
Serial No. : 09/971,086
Filed : October 5, 2001
Page : 3

Attorney's Docket No.: 06975-153001

REMARKS

Attached is a marked-up version of the changes being made by the current amendment.
Applicant asks that all claims be examined. Please apply any other charges or credits to Deposit
Account No. 06-1050.

Respectfully submitted,

Date: 12/20/2001



W. Karl Renner
Reg. No. 41,265

Fish & Richardson P.C.
601 Thirteenth Street, NW
Washington, DC 20005
Telephone: (202) 783-5070
Facsimile: (202) 783-2331

40072676.doc

09/971,086

Version with markings to show changes made

In the specification:

Paragraph beginning at page 8, line 1 has been amended as follows:

IProtocolSink is an interface for pluggable MIME filters that is used to receive notifications if/when data is available for filtering. IProtocolSink allows the APP to obtain a window handle for the client performing the bind operation. Although this window handle typically is used for other purposes, it can be used to determine whether an instance of a MIME filter originated from an internal browser [by examining the class name of the reported window handle].

Paragraph beginning at page 9, line 6 has been amended as follows:

If the data stream was requested by an instance of an internal browser, the registry structure is modified to include registrations of media players appropriate for internal instances of the browser or some other list of alternate media player registrations is accessed, and that modified or alternate registry is analyzed to determine which media player is registered for the MIME type (step 560). Then, an instance of an appropriate alternate media player is established (step 570) and the data stream is effectively routed for rendering to the alternate media player (step 580). Absent further processing, after the media stream is rendered, the MIME filter returns the data stream to URLMON. Therefore, to prevent invocation of a second media player according to the modified Windows[®] registry, it may be desirable or necessary to sink the data stream (step 590).

Paragraph beginning at page 9, line 16 has been amended as follows:

Fig. 6 illustrates one implementation for sinking a call for a default media player ordinarily generated for data streams of similar MIME type by external instances of a browser (step 590). In this implementation, the MIME filter changes the MIME type of the data stream returned to URLMON to a predetermined MIME type (step 610) for which no media player or a dummy media player has been registered. As a result, when the browser detects the altered MIME type in the data stream, the browser will initiate either no media player or an instance of this dummy media player, as appropriate (step 620). Then, to complete the data sink, the

modified data stream is effectively routed to this dummy media player, which [receives] accesses the data stream and either discards the data stream or performs some trivial or background function on the data stream.

Paragraph beginning at page 11, line 1 has been amended as follows:

The browser then initiates an instance of the alternate media player (step 770) and the data stream is effectively routed to it for rendering (step 780).

09/971,086